REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
JUNE 2009

2. REPORT TYPE
Technical Paper Post Print

3. DATES COVERED *(From - To)*
September 2008 – June 2009

4. TITLE AND SUBTITLE
MIXED-INITIATIVE PLANNING IN A DISTRIBUTED CASE-BASED REASONING SYSTEM

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER
In-House

5c. PROGRAM ELEMENT NUMBER
62702F

6. AUTHOR(S)
Kurt Lachevet

5d. PROJECT NUMBER
558S

5e. TASK NUMBER
IH

5f. WORK UNIT NUMBER
DP

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AFRL/RISC
525 Brooks Road
Rome NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/RISC
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TP-2009-11

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited. PA# 88ABW-2009-1201 Date Cleared: 9-June-2009

13. SUPPLEMENTARY NOTES
This paper was presented at the 14th (2009) International Command and Control Research and Technology Symposium June 15-17, 2009. Washington DC. This is a work of the United States Government and is not subject to copyright protection in the United States.

14. ABSTRACT
The USAF Command and Control (C2) is undergoing a transformation to enable a full-spectrum, joint warfighting capability. To be able to meet the future challenge of employing forces anywhere in the world in support of national security objectives, the USAF requires a highly synchronized, distributed planning and replanning capability that is flexible and agile enough to adapt to any level of conflict. Complex planning systems require the expertise and knowledge of both humans and computers during the planning process. By allowing computers and humans to contribute their individual expertise to discrete areas of the planning process, a synergy is formed which cannot compete with fully automated systems or human-only planning processes. This paper describes an approach to providing mixed-initiative interaction in a distributed, case-based planning system that is under development through an in-house program at the USAF Research Laboratory Information Directorate. This approach to mixed-initiative planning identifies methods to achieve this important synergy by leveraging existing technologies such as distributed blackboards, case-based reasoning, formal plan representations, multi-agent systems, as well as semantic technologies.

15. SUBJECT TERMS
Mixed-Initiative, Planning, Distributed, Blackboard, Case-Based Reasoning, Multi-Agent System

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	22	Kurt Lachevet
					19b. TELEPHONE NUMBER (*Include area code)*

14th ICCRTS

"C2 and Agility"

Mixed-Initiative Planning in a Distributed Case-Based Reasoning System

Topic 9: C2 Architectures and Technologies

Kurt Lachevet

Air Force Research Laboratory Information Directorate

Information Systems Research Branch (RISB)

525 Brooks Road

Rome, NY 13441-4505

kurt.lachevet@rl.af.mil

POSTPRINT

Mixed-Initiative Planning in a Distributed Case-Based Reasoning System

Abstract

The USAF Command and Control (C2) is undergoing a transformation to enable a full-spectrum, joint warfighting capability. To be able to meet the future challenge of employing forces anywhere in the world in support of national security objectives, the USAF requires a highly synchronized, distributed planning and replanning capability that is flexible and agile enough to adapt to any level of conflict. Complex planning systems require the expertise and knowledge of both humans and computers during the planning process. By allowing computers and humans to contribute their individual expertise to discrete areas of the planning process, a synergy is formed which cannot compete with fully automated systems or human-only planning processes.

This paper describes an approach to providing mixed-initiative interaction in a distributed, case-based planning system that is under development through an in-house program at the USAF Research Laboratory Information Directorate. This approach to mixed-initiative planning identifies methods to achieve this important synergy by leveraging existing technologies such as distributed blackboards, case-based reasoning, formal plan representations, multi-agent systems, as well as semantic technologies. This paper will also discuss the current prototype, current challenges, as well as future research and work planned towards a fully functional, mixed-initiative planning system.

Keywords: Mixed-Initiative, Planning, Distributed, Blackboard, Case-Based Reasoning, Multi-Agent System

1 Introduction

1.1 Problem Statement

The U.S. and other highly industrialized nations have developed military capabilities that excel in conventional force-on-force warfare, especially where tactics are well developed and known. However, modern adversaries have devised the strategy of not going “head-to-head” with these capabilities and instead combat modern conventional forces with unconventional tactics. One example of the result of a weapon system being vastly superior is the case of the air superiority fighter which modern adversaries totally avoid putting themselves in a position to contest them.

To meet these future challenges, U.S. forces are in the midst of a “transformation” to not only support traditional high-tempo, large force-on-force engagements, but also smaller-scale conflicts characterized by insurgency tactics and time-sensitive targets of opportunity. This transformation requires a vastly new Command and Control (C2) process that can adapt to the any level of conflict, provides a full-spectrum joint warfighting capability, and can rapidly handle any level of complexity and uncertainty.

1.2 Future C2 Requirements and Information Age C2 Solutions

To meet these future challenges, the U.S. Air Force (USAF) needs to move towards a model of continuous air operations not bounded by the traditional 24-hour Air Tasking Order (ATO) cycle. Meeting these objectives will require a highly synchronized, distributed planning and replanning capability. As a potential way ahead, in May 2006 released a revolutionary vision paper (Braun 2006) depicting what a potential future C2 environment could be. Four key concepts emerged from this vision of a future AOC:

- Distributed/Reachback planning
- Redundant/Backup planning
- Continuous planning
- Flexible, Scalable, Tailorable (Agile) C2

Experience with recent operations also reveals that the C2 process must transition from a process of observation and reaction to one of prediction and preemption. To achieve this, we will need to go beyond the focus of military operations, and instead address the entire spectrum of Political, Military, Economic, Social, Infrastructure, and Information (PMESII) features.

The focus of this research has been founded on two emerging concepts for the future of C2. Developing a C2 environment that supports the vision of Network Centric Operations (NCO) was task number one. The tenets of NCO are (Wells 2007):

- Information sharing
- Shared situational awareness
- Knowledge of commander's intent

1.3 Related Work

As we increase the use of technology in Command and Control (C2) processes, we must be extremely sensitive to the way in which humans interact with these systems in order to retain agility and avoid scripted, easily anticipated C2 methods. Extensive work has been and is underway in this area of human-computer interaction of which some will be outlined below.

A DARPA program, Mixed Initiative Control of Automa-teams (MICA), published a final technical report in July 2004. The MICA program researched the control of teams of unmanned platforms focusing on the following research areas:

- Team Composition and Tasking
- Team Dynamics and Tactics
- Cooperative Path Planning
- Uncertainty Management
- Variable Initiative Interaction

Although the project was cut short citing "evolving priorities within DARPA" (McDonnell 2004), there was significant progress made. A major difference between the research to be outlined in this paper and the MICA program is the scope. The MICA program was tightly scoped towards UAV control, whereas this research is based on planning on much more abstract level. The research area of Variable Initiative Interaction (VII) most closely relates to the scope

of this paper. The report defines the purpose of VII as to provide the user with "sufficient information to develop situational awareness necessary to interact with and control heterogeneous teams of UAVs" (McDonnell 2004). Although the MICA research was scoped to UAV control and planning, this research plans to build upon the idea of the machine providing situational awareness by allowing the machine to provide other information beyond decision-aids, to include actual planning decisions and suggestions through the use of expert systems.

A paper titled *Expectation Failure as a Basis for Agent-Based Model Diagnosis and Mixed Initiative Model Adaptation during Anomalous Plan Execution (*Mulvehill 2007) outlines an algorithm and approach to improving models in which upwards of a thousand plans may be concurrently running. A mixed-initiative approach is taken for the improvement of the model. Their algorithm asses many 'anomalies' in the model, and determines a set of suggested adjustments to be provided to the user. Only the top ranked proposed changes are then presented to the user. Filtering results in this way is a very common method of mixed-initiative interaction within a system. This idea of proposing top ranked suggestions to the user will also be discussed, among other ways of mixed-initiative interaction, in the approach outlined in this paper.

Other work by Thomas Sheridan at Massachusetts Institute of Technology includes classifying automation levels of certain aspects in a system. One paper (Sheridan 2000) provides a "1-10" scale for levels of automation across four areas of system functionality. This work is extremely useful for categorizing and comparing levels of autonomy in a given system as well as providing a numerical metric.

1.4 Objective of this Approach

The objective of this approach to Mixed-Initiative Planning is to place the human in the loop of the previously highly-autonomous system. It has always been known that the human would need to be tightly intertwined in the loop during a plan's development; however, focus has previously been placed on the development of the architecture and its supporting technologies.

The objective of this approach is part of the long-term goal of the Distributed Episodic Exploratory Planning (DEEP) project which is to develop in-house a prototype system for distributed, mixed-initiative planning that improves decision-making by applying analogical reasoning over an experience base. Carbonell (1983) explains how analogical reasoning is a "powerful mechanism for exploiting past experience in planning and problem solving." The two key objectives of DEEP are:

- Provide a mixed-initiative planning environment where human expertise is captured and developed, then adapted and provided by a machine to augment human intuition and creativity.
- Support distributed planners in multiple cooperating command centers to conduct distributed and collaborative planning.

The architecture of DEEP was explicitly designed to support these tenets of NCO in a true distributed manner. Because DEEP is not based on any current C2 system, we are able to explore concepts such as combining planning and execution to support dynamic replanning, machine-mediated self synchronization of distributed planners, and experiment with the impact of trust in an NCO environment (i.e. "Good ideas are more important than their source").

1.5 Research Areas

Alberts and Hayes (2007) advocate bold new approaches beyond current organizational process, focusing on what is possible for NCO. High priority basic research topics recommended as areas to systematically explore are:

1. Taxonomy for planning and plans;
2. Quality metrics for planning and plans;
3. Factors that influence planning quality;
4. Factors that influence plan quality;
5. Impact of planning and plan quality on operations;
6. Methods and tools for planning; and
7. Plan visualization

Pursuant to achieving the vision of DEEP, essentially all the above topics needed to be addressed. The first topic was the starting point and has received the most attention. The earliest effort in support of distributed planning was on CPR, an object-oriented plan framework developed under the ARPA-Rome Laboratory Planning Initiative (ARPI). CPR is based on the Unified Modeling Language (UML) which is well suited as the human-machine dialog to support mixed-initiative planning. The recursive nature of CPR supports multi-level planning at all levels (strategic, operational, and tactical), along with plan fragments supporting distributed planning on a plan simultaneously. Along with the work outlined in this paper, another current research topic for DEEP is maintaining referential integrity when distributed planners simultaneously work on multiple sub-plans and/or plan fragments of a larger plan.

1.6 DEEP and Agile C2

The DEEP architecture was designed to be flexible & agile enough to adapt to any level of conflict and any type of situation. This agility arises from the ability of the case-based reasoning system to apply past experiences to current situations, whether they are very similar or seemingly irrelevant. Consider the current DEEP demo described in Carozzoni, et al. (2008) which applies a World War II case base to a Humanitarian Response Situation. At first glance it may not appear that these two situations have little in common; however, the system was agile enough to adapt the logistics planning from the WWII cases to the current situation.

This approach moves towards a more agile Command and Control model by addressing one of the requirements listed by Alberts (2007):

> *Agile C2 requires a "rich and continuous set of interactions between and among participants…and with the broadest distribution of decision rights."*

The current DEEP implementation lacks this *continuous set of interactions* between the contributors in the planning system. Currently, a planner sits at the console, enters a set of inputs and clicks 'RUN'. There are no opportunities for the human to intervene or collaborate with other humans. The approach outlined in this paper aims to improve the DEEP architecture by introducing these important *continuous sets of interactions.*

2 Approach

This Mixed-Initiative Planning approach aims to maximize the synergy between humans and machines. Not only will both humans and computers be contributing their expertise to the plan under development, but the computer will also be providing pertinent information to aid the human in their decision making processes.

As with all semi-autonomous systems, a major challenge is determining which functions should be automated. Sheridan (2000) proposes a framework that organizes system functionality into four classes:

- Information Acquisition
- Information Analysis
- Decision and Action Selection
- Action Implementation

Although this breakdown is admittedly simplistic, it allows us to organize the functions and determine their level of autonomy. Because this is a planning system, the actual Action Implementation would be the execution of the plan. Therefore, this system's current scope falls into the first three functionality classes. Something also worth noting is that this system aims to support *adjustable autonomy*, meaning that in some instances the user may want to make decisions, whereas in other instances they may wish for the machine to run through unconstrained.

Fundamentally, when a human enters information into the system, it is analyzed, and enhanced with information by the machine, and feedback of this information entered is presented to the user, to help steer the plan development process. The technologies, methods, and proposed architecture which provide this important synergy will be discussed in the following sections.

2.1 High Level Conceptual Overview

Figure 1 depicts the conceptual interaction of humans and computers contributing to an evolving plan. In the diagram, there is the circle on the left which contains the human entities in the system and a circle on the right which contains the computer entities in the system. The arrows from the entities to the evolving plan depict each entities contribution towards the plan. The arrows from the machine side to the human side depict the information that the machines provide to the users to help make their decision.

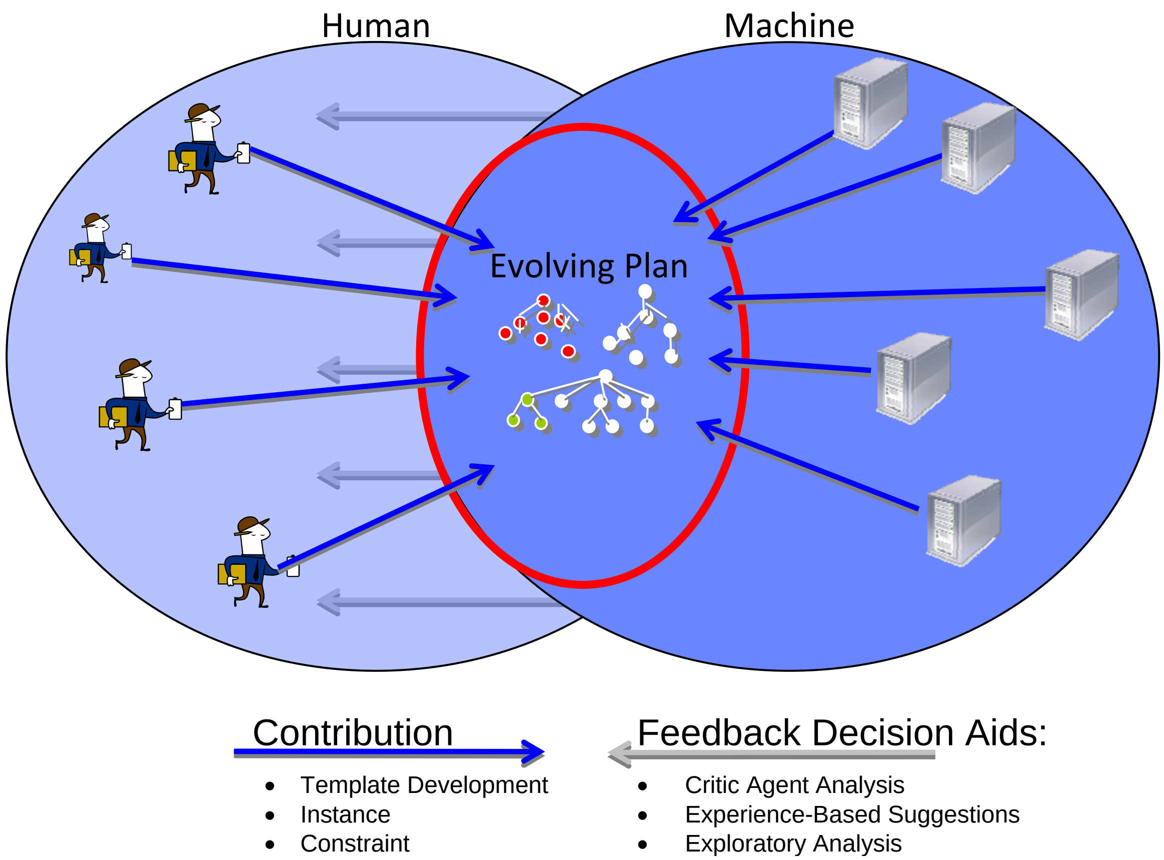

Figure 1 – Human / Machine Interaction

Contributions to the plan can take several forms and originate from both human and machine entities. These contributions may take the form of template development, instances of a template, or part of a template. For example, there may be an existing or generated template which abstractly defines a plan to “Deliver Equipment”. A detailed discussion of what is meant by templates will be discussed in the section describing ways of ‘Human Interaction’.

This approach also allows placing positive or negative constraints on a plan. A constraint may take many forms, from enemy position to logistical constraints. For example, an entity, be it human or machine, may know that there is a shortage of a commonly used supply and add it as a negative constraint. On the other hand, there may be a requirement passed down to utilize a certain asset and added to the plan in the form of a positive constraint.

As presented earlier, this approach involves both human and machine entities bringing their individual expertise as a contribution to the planning process. For example, a software entity may be an expert at providing certain weather data and incorporating it into the plan, while a human logistician can offer their knowledge at their level of planning expertise. On a different level, a strategic planner would supply their expertise at a higher level. Figure 2 illustrates this concept of distributed entities contributing at difference levels with their individual expertise. The graphs in the center of the diagram depict the plan representation being used. Because an individual is

competent in their area, clear boundaries between planning levels do not need to be defined by this approach, and can be simply incorporated and used as defined by current planning doctrine.

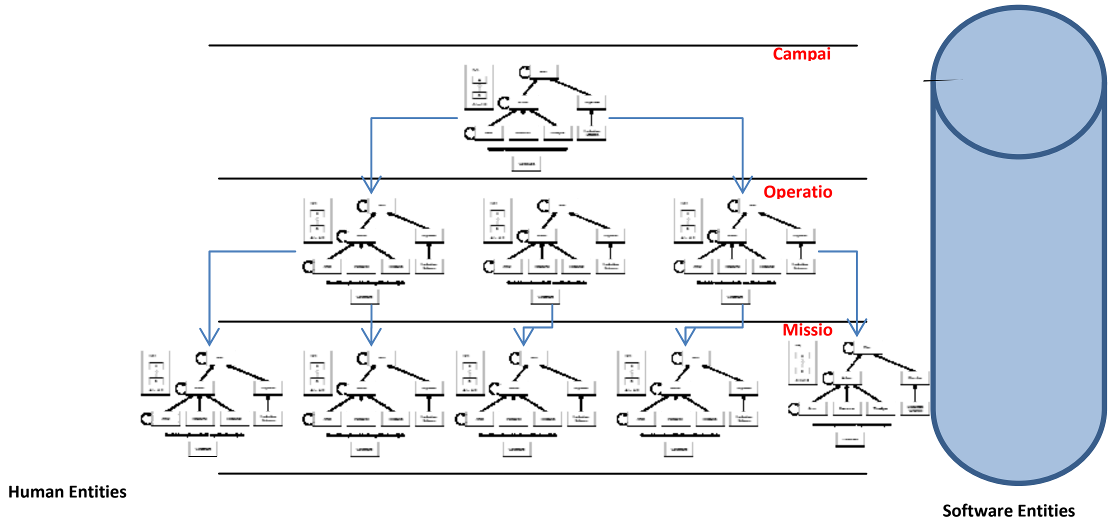

Figure 2 – Multi-Level Planning

Not pictured in Figure 2, is the valuable information that machine entities provide back to the user as can be seen in Figure 1. This information may include plan scoring, warnings, suggestions, or other information that may steer the user into the development of a better plan.

As can be seen, the user will be highly integrated into the planning process, and attempting to move away from the discrete, synchronous interaction between the human and machine. To move towards this more continuous, asynchronous human-machine planning process, a number of technologies must work in concert. In the next sections, a proposed architecture and user-interface mockup which will provide these conceptual capabilities will be presented.

2.2 Components & DEEP Architecture

DEEP is a system-of-systems architecture (Figure 3), comprised of the following systems:

- Distributed Blackboard for multi-agent, non-deterministic, opportunistic reasoning
- Case-Based Reasoning system to capture experiences (successes and/or failures)
- Episodic Memory for powerful analogical reasoning
- Multi-Agent System for mixed initiative planning
- ARPI Core Plan Representation for human-to-machine common dialog
- Constructive Simulation for exploration of possible future states

Consider the DEEP architecture depicted in Figure 3. The starting point for entry into the system occurs when a commander describes a new mission using a *planning agent* (1). The planning agent allows for the commander to input information into the system which defines their *current objectives*. These objectives, along with other information, such as resources, locations, and time constraints, are collectively known as the *situation*. This situation is then placed on the shared blackboard (2). The blackboard would in turn notify all registered systems of the existence of the

new situation. Using the given situation, the other planning agents, with their associated case bases and cased-based reasoning capabilities, would each search their case base for relevant past experiences (3). These results are then modified to fit the current situation (4) and are posted to the blackboard as *candidate plans* (5). Once the candidate plans are on the blackboard, they are adapted by specialized *adaptation agents* to further refine these plans to meet the current situation (6). These plans are now ready to be critiqued by the *critic agents*.

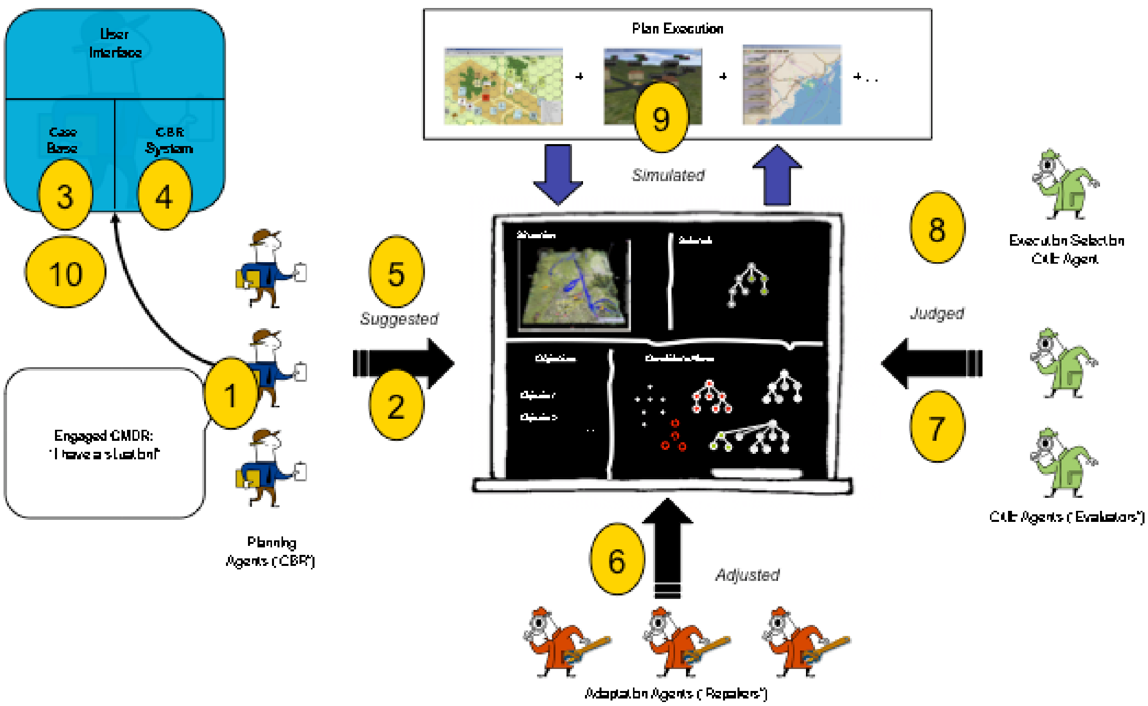

Figure 3 - DEEP Architecture

Critic agents concurrently scrutinize the candidate plans and score them based on their individual expertise (7). Once the plans are scored, the *execution selection critic* gathers the adapted plans along with their scores, determines their overall scores, and selects a number of top rated plans to be executed (8). The top rated plans are now executed (currently in a simulated environment) (9). Once a plan completes execution, the results are combined with the plan and assimilated back into the original planning agent's case base (10).

Although we have described this planning and execution as a single flow through the system, in reality few plans will execute without changes. The DEEP architecture supports the modification of currently executing plans through feedback of partial results of plan execution into the blackboard. This allows the plans to be run through the adaptation and critique processes as many times as needed.

The DEEP architecture also includes a messaging system, various knowledge objects, a shared data storage system, along with a number of agents, all described later in this chapter. For convenience, we will describe the pieces in the architecture in the order in which they might be typically used. One should bear in mind, however, that in this type of mixed-initiative system, there will rarely be a clean path from the initial planning problem to the final solution.

2.2.1 Plan Representation

The various DEEP systems all use a common knowledge representation to facilitate their interactions. We know that the future of military planning is not just for the Air Force, but rather will involve participants from various agencies (both military and civilian), possibly planning at different levels of abstraction. Thus, DEEP was designed to support plans for joint, coalition, and civilian operations as well handle plans at different abstraction levels (i.e., strategic, tactical, or operational). Planning for heterogeneous operations also means that the plan representation has to be able to consider the semantics of terms used in the plan, ensuring agreement among all participants. This is an ongoing research topic, discussed in detail in a later section. Finally, because DEEP is a mixed-initiative environment, the chosen plan representation must be easily machine-readable as well as presentable to a user.

The ARPA-Rome Laboratory Planning Initiative (ARPI) conducted research on several plan representations. The culmination of that effort was the Core Plan Representation (CPR), shown in Figure 4. Although there are many projects and efforts underway to develop interoperable standards, CPR was selected for DEEP as best meeting the above criteria. Because of its flexibility, it can easily map into other languages. For example, DEEP team members quickly created a mapping from its programmatic CPR structure into an XML file for data persistence and storage. CPR is also an object-oriented structure that is agnostic to the planning abstraction level (i.e., strategic, tactical, or operational) (Pease A. , 1998). Its natural object oriented structure also lines up very well with the machine reasoning capability DEEP requires. The original CPR structure (Figure 4) has been adapted to meet the needs of DEEP as they have evolved over time.

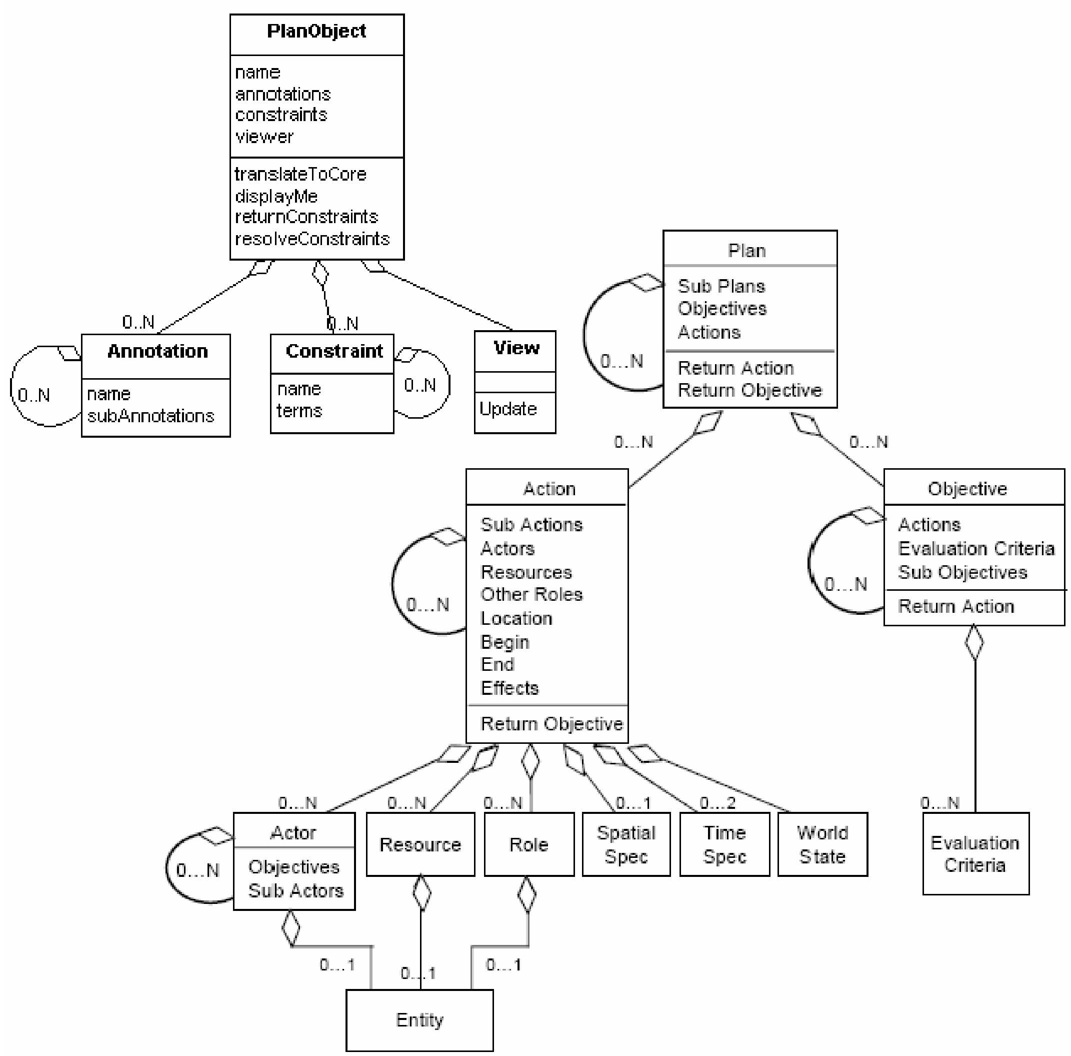

Figure 4 - Core Plan Representation

In DEEP, CPR is used to represent individual experiences, or *cases*, which are composed of a plan, events, and one or more outcomes. The attributes of the plan are used by the cased-based reasoning system (Section 2.2.3) to determine the similarity of past cases with the current situation. Execution (currently through simulation) of the plan populates the events and outcome sections. DEEP-CPR was extended from the base structure shown in Figure 4 to support a much deeper reasoning capability of plans.

2.2.2 System Messaging

CPR is the foundation for the DEEP architecture and used by all components, thus a formalized messaging model is required for the interactions within the systems. The systems that interact with one another include various types of agents along with the system blackboard. To accomplish this, a formalized messaging scheme based on inter-agent communication is required with a defined structure so that new systems are able to understand incoming messages as well as

transmit their own. The DEEP architecture includes a formal messaging scheme to be used by the other systems.

In the current DEEP architecture, the communication protocol used is the publish-subscribe communication paradigm using the blackboard (Section 2.2.4) as a medium. At a high level, systems subscribe to the blackboard and are notified when new information is added. Because of the push to create a functional proof-of-concept architecture, a simple taxonomy is currently in place to determine notification and message types until a more formalized communications protocol is established. The blackboard mediates all messaging using its defined messaging scheme and connectivity medium. To prohibit the distributed planning aspect of DEEP deteriorating to "chat-room" type collaboration, an artificial barrier has been placed on human-to-human direct planning. Therefore, agents of any kind (human or software) do not communicate with each other directly, but instead use the blackboard as a hub of communication. For example, consider the mixed initiative scenario where a critic agent requires input from a user. To obtain this input, the critic agent would send a message through the blackboard to the appropriate interface agent; the reply would similarly be routed back through the blackboard.

2.2.3 Distributed Case Based Reasoning System

DEEP currently uses Case-Based Reasoning (CBR) as the experience based reasoning system. Figure 5 illustrates how the CBR cycle is applied to case-based planning used in the DEEP architecture. Case-based planning makes use of past experiences to implement new plans and retain their outcomes. That is, "Case-based planning is the idea of planning as remembering" (Hammond, 1990). The planning agent allows the user of the system to input a situation using a user interface. Once the operator feels comfortable with the input, the agent, via the user interface, allows the situation to be forwarded to the blackboard. The situation includes statements about the problem's objective, locations, actors, resources, and times. While interacting with the user interface, the operator can also interact with plans on the blackboard and view the case base associated with the planning agent. The case base for each planning agent will be unique.

Once a situation has been placed on the blackboard, the blackboard will broadcast a message notifying all registered systems about the new problem. The listeners in the other planning agents determine what type of object was placed on the blackboard, and react to a new situation by initiating cased-based reasoning for the new problem. See Ford & Carozzoni (2007) for a complete explanation of the CBR process used in DEEP. The CBR process selects the best set of cases from its case base and posts them onto the blackboard as candidate plans. Once the candidate plans are placed on the blackboard, they are processed by the critic agents (discussed in detail below in Section 2.2.6).

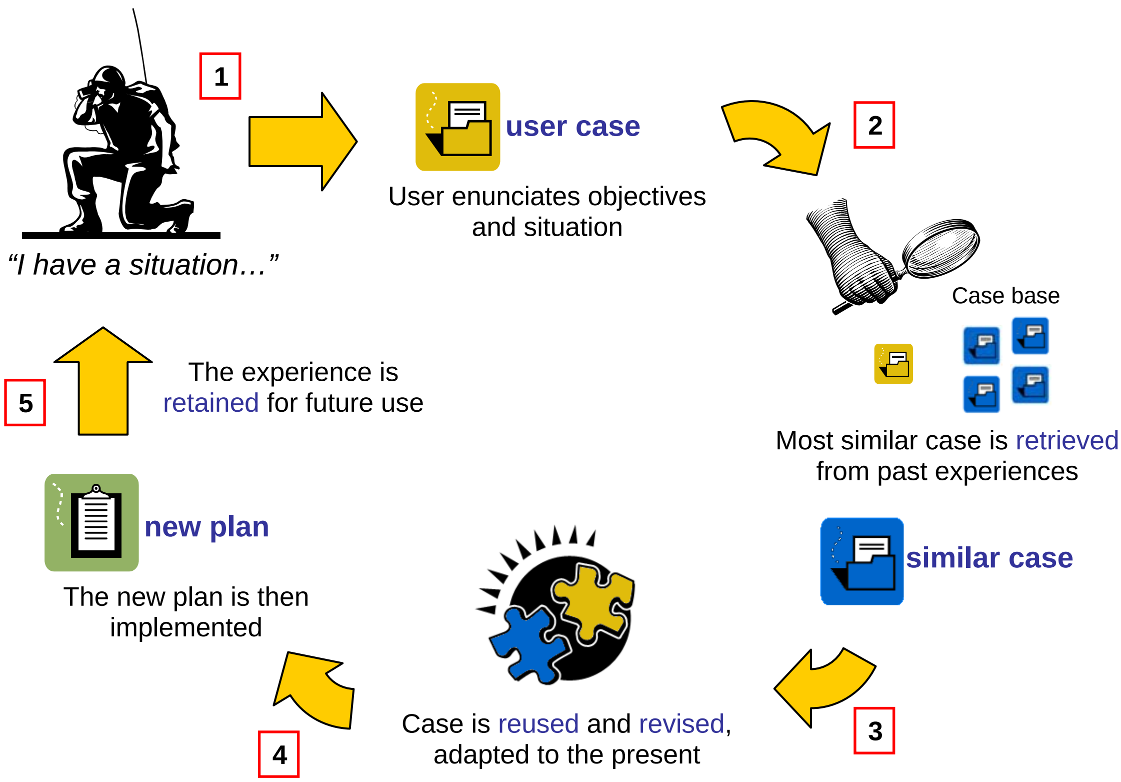

Figure 5 - Case-based Planning

Each planning agent is expected to have a unique case base, since each planning agent represents the experience of some entity or group of entities. The case base of an entity can contain experiences of any kind. This variety is readily supported by DEEP's plan representation, CPR, because of its ability to work with planning knowledge at different levels of abstraction.

The interface/planning agent is indeed a multi-faceted entity, providing an interface to the user, an interface to a case base, and an interface to a reasoning engine. These interfaces are important due to tight interaction of these systems. Little processing is done by the planning agent itself, but rather by an external system that it interfaces with (e.g., CBR System). The agent itself is the medium between the reasoning process and the blackboard as well as the human and the blackboard. Now that the plans are on the blackboard and ready for evaluation, it is time to discuss the critic agents.

Current work also involves taking parts of disparate experiences and determining which parts may be joined together in order to form a coherent plan. This work is critical to the overall functionality of this approach to mixed-initiative planning.

2.2.4 Distributed Blackboard

As can be seen from the DEEP architecture in Figure 3, the various DEEP systems rely on a shared knowledge structure to act as a medium of communication and interaction. Also, in order to support the NCO vision discussed in Section 1.2, the DEEP architecture requires a mechanism that supports reach-back in a distributed system. A blackboard system was chosen to fulfill this

need as it not only functions as a shared memory for the DEEP system, but as we discussed in Section 2.2, it provides other functionality as well.

A blackboard system is an opportunistic artificial intelligence application based on the blackboard architectural software engineering paradigm (Corkill, 1991).The blackboard system functions as a central knowledge store facilitating communication and interaction between the different software systems, including interface agents, critic agents, and simulation engines (explained later in this chapter). These interactions are made possible by the sharing and passing of objects.

To fully meet the requirements of the DEEP vision for distributed C2, a *distributed* blackboard system was required. Current commercial and open source blackboard system implementations are not distributed, so the paradigm needed to be extended from a monolithic to a distributed environment. The current DEEP blackboard was designed and implemented using constructs to enable a true distributed, shared memory.

Traditionally, a blackboard consists of three discrete components: the blackboard knowledge structure which is a central repository for knowledge objects, the knowledge sources which are specialist software modules (agents in the DEEP software architecture) that provide specific expertise required by the system, and a control component which controls the flow of objects and problem-solving activity in the system (Corkill, 1991).

The Distributed Blackboard System is still under development and is currently moving towards a distributed database as a persistent storage mechanism.

2.2.5 Adaptation Agents

Adaptation agents in the DEEP architecture are software agents that specialize in further refining a plan based on their particular area of expertise. As explained earlier, the initial plan that is instantiated to the new situation and placed onto the blackboard is a "rough cut" and needs supplementary revision. When the adaptation critic agent receives notification from the blackboard that there is a new instantiated plan on the blackboard, it reviews the plan, makes its changes, and posts a new version of the plan with its adaptations.

There are many possibilities for different adaptation agent specializations. For a proof-of-concept, a Capabilities Adaptation Agent was designed, developed, and integrated into the DEEP architecture. The Capabilities Adaptation Agent's specialization is validating that the actors in the instantiated plan are capable of performing the actions to which they were assigned. In order to accomplish this, it first has to be determined what roles an actor is capable of performing and validate that it is consistent with the action to which it has been assigned. Actors in the DEEP architecture have both default roles as well as specialized roles for the situation that need to be taken into account. If a given actor is not capable of the role that has been assigned, a new actor must be found to replace it. The agent looks in the current situation for similar available actors, where similarity is determined by traversing an actor/role taxonomy and selecting one with a minimal semantic distance (Ford & Carozzoni, 2007). This new, similar actor then replaces the incapable actor in the action. After the agent has adapted this plan using its specialized knowledge, it then posts the updated plan to the blackboard.

The use of Adaptation Agents is also an area where mixed-initiative interaction can once again be brought into the system. Interfaces could be made to allow the plan to be displayed to a user, either as a whole or in a specific way, which would allow the user to apply his or her expertise and adapt part of the plan.

In the process discussed so far, a scenario has been created by specifying objectives, methods and resources that were then placed on the blackboard. Each planning agent has examined its case base to find similar situations and used them to instantiate a new plan based on its past experience. Now that the plans on the blackboard have been instantiated, and refined by multiple Adaptation Agents, they are ready to be scored and criticized by the Critique Agents.

2.2.6 Critique Agents

This category of agents can be quite extensive, but the present DEEP system only uses one for demonstration purposes. The particular critique agent implemented is a weather agent, but the future possibilities include political, logistics, ethical, legal and cyber agents, among others.

Scoring agents focus themselves on certain areas of a plan. A weather agent, for instance, would focus on how weather impacts the plan and ignores other areas such as political fallout. A legal agent would not focus on weather, but instead would focus on the legal aspects of the plan. These agents will find and use relevant data and ignore data that is of no concern to them. Critique agents do not change the plan as adaptation agents do; rather, they only analyze how the plan may work in the particular subject area. To have a subject area of expertise, these agents usually wrap or communicate with an outside knowledge source that specializes in that area. The weather agent for example has a weather feed it can communicate with, an understanding of weather rules and the weather capabilities of actors in the plan.

During the DEEP process, critique agents use the adapted plans on the blackboard for evaluation. The agents will use the data they need in the plan to further their processing. For example the weather agent will extract the location data contained in the plan and then use that location data to gain weather information using an external source, such as an RSS (Really Simple Syndication) feed. Once the CPR plan object has been parsed for the needed data, the critique agent will process it. The implementation behind evaluation will be different for each critique agent as will the data required for them out of CPR. It is possible as new critique agents are added CPR will need to evolve to include more information. Once evaluation has finished, these agents use a scoring algorithm to produce a score that is tied to that particular plan, which is posted to the blackboard.

These agents follow a general technical scheme. They implement the Java Agent Development (JADE) Framework, register to and setup listeners to the blackboard, have a knowledge source either internally or externally, have an evaluation implementation, and scoring algorithm. These agents can also use human sources as their knowledge base allowing for mixed-initiative interaction.

2.3 Mockup

First, it must be stressed that this screenshot is a mockup, and is not a functional implementation. This objective of this mockup is to propose a possible user interface for use as a front-end on the Distributed Episodic Exploratory Planning (DEEP) project. The mockup will be used to discuss

the different capabilities that will be provided through this mixed-initiative planning approach, allowing the user to interact with the mixed-initiative system.

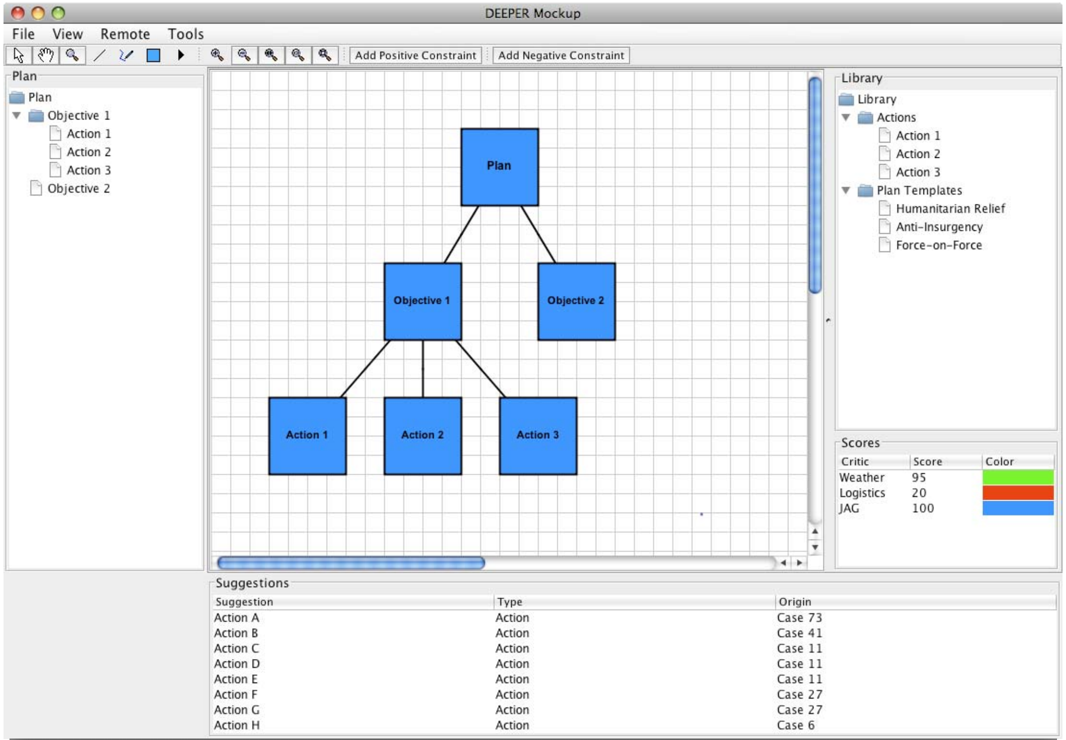

Figure 6 – Mockup User Interface

This application window displays a number of inner windows which allow the user to interact with the developing plan, as well as view important information provided by the other software entities in the system. As discussed earlier, this is a distributed system, so we can think of this application window as a view into the planning system, and that there is many-to-one mapping of these application windows into the one planning system used to develop the plan. A challenge here is to provide an interface which is tailored to a wide range of users with different planning expertise. For instance, a strategic level planner will want to see different information than a logistician. One avenue of future work includes research and testing into finding the optimal technique to providing information in the correct context for the user's level of expertise.

Since a plan is under development, there must be one or more ways to interact with the evolving plan. This mockup proposes two views of the plan, both of which offer ways of interacting with the plan: a Tree view and a Graph view. The Tree view, shown in Figure 6, shows an overview of the plan. This view of the plan is more of an overview, and will not provide most of the functionality that the Graph view does.

The Graph view shown in Figure 6 will be the main focus of the user. This will show a graphical view of the plan under development, and allow a multi-level view of the plan. Functions such as Zooming, and Collapsing of nodes will allow a user to adjust the view to their level of

preference. For example, a high-level planner might choose an option to collapse the lower-level planning details so they can focus on their level of expertise.

2.3.1 User Interactions

A user may use the interface to contribute to the evolving plan in several ways. These include:

> *Template Specification* – This is a high level way of placing a plan specification. This contribution would allow the user to say that they expect to have a certain plan outline that will be filled in as the plan evolves. For example, a user may specify that their specific objective will have three distinct actions, or phases as a part of the overall plan.
>
> *Instance Specification* – This capability allows a user to specify a more specific piece of the plan than the *template specification*. In this case, instead of specifying that the objective has three actions, they would specify an instance of each action, for example neutralizing an enemy radar site.
>
> *Drag and* Drop – This allows a user to drag and drop plan fragments from previous experiences into the plan under development. The specifics of the plan would then be adapted to the current situation by the underlying reasoning engine. For example, a planner may know that in order to neutralize the enemy radar site, they first must find and identify enemy radar sites through surveillance and reconnaissance. Since this is a commonly used action, a template to perform general Intelligence, Surveillance and Reconnaissance actions could be reused from previous plans.
>
> *Positive/Negative Constraints* – This allows a user to specify specific positive or negative constraints for the plan under development. These constraints could be placed on any part of the plan to ensure that a specific detail will remain intact throughout the planning process. An example of a positive constraint would be specifying that a specific asset must be used as a part of the plan. Shown in Figure 6 are the positive and negative constraint tabs that would allow the user to specify their constraints.

2.3.2 System Feedback – Scoring and Suggestions

The user will also be receiving a wide variety of information regarding the evolving plan. Looking back at Figure 6, you'll notice two windows labeled *Suggestions* and *Scores*. Along with the information provided in these windows, will be other feedback from the system such as alerts, warnings, and other analysis. The information provided to the user will be discussed below.

The Scores window provides useful feedback regarding the scores the plan has received from the various *Critic Agents* present in the system. These *Critic Agents* are software components that score the plans and provide these scores back to the system. *Critic Agents* will be further discussed later in the paper.

The representation of a plan score is a current area of research in the DEEP project. Part of this research is how to represent a score to the user. This score may take the form of an absolute number, relative number to other scores, or perhaps even a color. Consider a table of scores with

a respective color next to their scores. As the plan is scored better, it may have a green color next to the respective scorers, whereas a poor plan may display a red color.

Suggestions are presented to the user based on the experience-based reasoning system. More on the specific of this system will be discussed in a later section. On a high level, this system will use its previous experiences to supply suggestions to the user. This system takes into account the current situation, or context, along with the available resources. Based on the objectives the user is specifying, this *Suggestions* capability will take these objectives into account with the context, and come up with creative analogies based on past experiences and present them to the user. For example, if a user enters a high-level, strategic Objective, the underlying reasoning system will take the context of the current situation into account and present a possible solution or solutions, based on past experiences. This solution may range from a small part of the plan to a whole plan. Depending on the level of desired autonomy, these suggestions may be automatically incorporated into the plan, or only supply a queue to the user to help steer the development of the evolving plan.

The previous sections have presented a high level overview of the human and machine interaction in this distributed, experienced based system. It also provides an example implementation of exactly what the human might expect to see in an application setting. Also discussed was the architecture of technologies used to enable these capabilities which are under development in a project underway at the Air Force Research Laboratory Information Directorate.

3 Experiment / Metrics

Plans are currently in place for the implementation of this approach over the summer of 2009. Upon completion of this implementation, the analysis of this implementation will be measured through two stages of experiments:

1. Measure of Effectiveness
2. Measure of Performance

The first experiments will be aimed at determining the measures of effectiveness of this mixed-initiative planning approach. This set of experiments will act as a 'gate' to the second set of experiments. If this approach does not meet a specific level of effectiveness, the shortfalls will have to be identified, documented and determined if they can be tractably addressed. The hypotheses in this set of experiments will measure the levels of autonomy using the metrics proposed by Sheridan (2000). Given the information presented, it is expected that the level of human interaction with the system will increase significantly.

Assuming that the level of effectiveness of the system is at a satisfactory level, the next set of experiments will compare this approach's performance with other approaches to mixed-initiative planning. The experiments will compare and analyze multiple approaches using various metrics for various aspects of the planning system, including a comparison using performance metrics of the old system to the new system.

4 Future

Now that an approach has been outlined, the first work will include demonstrating a proof-of-concept implementation. Once this prototype is functional, experiments will be conducted to determine the effectiveness of the implementation by measuring the metrics defined.

If the measures of effectiveness indicate that the approach is a satisfactory solution to the problem, then the process of improving the approach may begin. The measures of performance will be used as a baseline to further experiment with methods of improving the performance of the Mixed-Initiative Planning system. Some avenues of research on improving performance include:

- Plan Visualizations – Efficiently presenting a plan to the user is an area currently under research. Current research in the cognitive aspects of human-computer interfaces should be leveraged and experimented with to allow optimum plan visualization and interaction with the user.
- Adjustable Autonomy – Adjustable Autonomy as a concept briefly mentioned earlier, however further research could be conducted to determine the correct level of machine autonomy.
- Commanders Intent – Future research should include the capturing of intent to support power-to-the-edge principles.
- Situation Editor – This user interface could also encompass a situation editor to add, remove, and modify pieces of the current situation.
- Tailored Information – Future work must include research and testing into finding the optimal technique to providing information in the correct context for the user's level of expertise.
- Other Human Entry Points – Great interest has been shown in providing means for the human to have a view into the 'black-boxes' in the system. This interface could also be a stepping stone towards achieving this goal.

As you can see, the success of this approach could be a stepping stone and research platform for future research paths. Most importantly, a prototype of this approach would provide a functional user interface for the DEEP prototype architecture.

5 Conclusion

This paper has outlined an approach to Mixed-Initiative Planning using technologies being developed under the DEEP project. This approach hopes to enhance DEEP's agility by providing users a functional way to interact with an adjustably-autonomous system providing the following capabilities:

- Level-Independent Mixed-Initiative Planning – Allow experts, both human and machine, to contribute and their level of planning expertise.
- Asynchronous Mixed-Initiative Planning – Allow the parallel, asynchronous planning between the human and machine as opposed to a scripted, turn-based human-computer interaction.

- Plan Oriented Machine Interaction – Allow the machine to opportunistically contribute to the evolution of the plan.

Currently, DEEP demonstrates the power of analogical reasoning in using the historical, military-focused Guadalcanal experience-base to plan for a modern humanitarian relief operation. By implementing this approach which will allow tight interaction between users and machines, the DEEP system will become even more agile. DEEP will be used to experiment with the capability of analogical reasoning to improved planning speed, plan quality, and plan creativity, and plan agility, with the vision of becoming a truly distributed mixed-initiative planning capability in the future.

6 References

Alberts, D., & Hayes, E. (2007). *Planning: Complex Endeavors.* Command and Control Research Program.

Braun, G., et al. (2006). *AFFOR Command and Control Enabling Concept-Change. 2.* AF/A5XS Internal Report.

Carbonell, Jaime G. (1983). *Learning by analogy: Formulating and generalizing plans from past experience.* Machine Learning, An Artificial Intelligence Approach, Palo Alto, CA, 1983. Tioga Press.

Corkill, D. (1990). Blackboard architectures and control applications. *Proceedings 5th IEEE International Symposium on Intelligent Control* . Piscataway, NJ: IEEE.

Corkill, D. (1991, September). Blackboard Systems. *AI Expert* , *6* (9), pp. 36-38.

Corkill, D. (2003). Collaborating software: Blackboard and Multi-Agent Systems & the Future. *Proceedings of the International Lisp Conference.* New York.

Carozzoni, J.; et al. (2008). *DEEP: Distributed Episodic Exploratory Planning*. Interim Technical Report. Rome, NY.

DeStefano, C., Lachevet, K., & Carozzoni, J. (2008). Distributed Planning in a Mixed-Initiative Environment. *Proceedings of the 13th International Command and Control Research and Technology Symposium.*

Ford, A., & Carozzoni, J. (2007). Creating and Capturing Expertise in Mixed-Initiative Planning. *Proceedings of the 12th International Command and Control Research and Technology Symposium.*

Ford, A., & Lawton, J. (2008). Synthesizing Disparate Experiences in Episodic Planning. *Proceedings of the 13th International Command and Control Research and Technology Symposium.*

Pease, A., & Carrico, T. (1997). The JTF ATD Core Plan Representation: A Progress Report. *Proceedings of the AAAI Spring Symposium on Ontological Engineering.* . AAAI.

McDonnell Douglas Corp., (2004). *Mixed Initiative Control of Automa-Teams, Open Experimentation Platform*. Sponsored by Defense Advanced Research Projects Agency, Final Technical Report. AFRL-IF-RS-TR-2004-198.

Mulvehill, A. et al. (2007). Expectation Failure as a Basis for Agent-Based Model Diagnosis and Mixed Initiative Model Adaptation during Anomalous Plan Execution. *Proceedings of the 2007 International Joint Conferences on Artificial Intelligence.*

Sheridan, T. B., Parasuraman, R. (2000). A Model of Types and Levels of Human Interaction with Automation. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, Vol. 30, No. 3, May 2000.*

Wiederhold, G. (1994). Interoperation, Mediation and Ontologies. *Proceedings of International Symposium on Fifth Generation Computer Systems (FGCS94.* Tokyo, Japan.

Wilson, C. (2007). *Network Centric Operations: Background and Oversight Issues for Congress*. CRS Report for Congress.